

UNITED STATES PATENT APPLICATION FOR:

METHOD AND APPARATUS FOR PLACING SENSORS USING 3D MODELS

INVENTORS:

AYDIN ARPA

KEITH HANNA

SUPUN SAMARASEKERA

RAKESH KUMAR

HARPREET SAWHNEY

ATTORNEY DOCKET NUMBER: SAR 14953

CERTIFICATION OF MAILING UNDER 37 C.F.R. 1.10

I hereby certify that this New Application and the documents referred to as enclosed therein are being deposited with the United States Postal Service on 2/13/04, in an envelope marked as "Express Mail Service", Mailing Label **EU 137712965 US** addressed to: Stop PATENT APPLICATION, P.O. Box 1450, Alexandria, VA 22313-1450

Kimberly Boyd
Signature
Kimberly Boyd
Name
2/13/04
Date of signature

MOSER, PATTERSON & SHERIDAN LLP
595 Shrewsbury Ave.
Shrewsbury, New Jersey 07702
(732) 530-9404

METHOD AND APPARATUS FOR PLACING SENSORS USING 3D MODELS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims benefit of United States provisional patent application serial number 60/484,138, filed July 1, 2003, which is herein incorporated by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] Embodiments of the present invention generally relate to surveillance systems and, more particularly, relates to a method and apparatus for dynamically configuring sensor placement using a 3D model.

Description of the Related Art

[0003] The current preferred method for optimizing the design of a sensor (e.g., a camera) layout is to begin with a 2D plan view of the environment, and to place cameras manually on the 2D plan view. In many cases, the design finishes at this point. In other cases, the desired field of views of the cameras are computed using a spreadsheet that computes the pixel size of an object given a camera's field of view using a 2D model of the environment. The cameras are then positioned at the computed distance apart (from one another) in an attempt to make sure that objects in the camera view are the correct size, and that all of the desired portions of the scene are covered by a camera view.

[0004] One problem with this approach is that the 2D model does not accurately capture the complex 3D projection of the scene into the 2D imager. For example, a camera looking obliquely at the scene will cover a different portion of area on the ground compared to a camera looking less obliquely. This significantly changes the area of coverage, and makes it difficult to guarantee that an area of ground will in fact be covered when the cameras are actually installed.

[0005] Another problem associated with this approach is that regions of occlusion are typically missed, resulting in unexpected blind-spots in camera

coverage that are only apparent when the cameras are installed. It is often too expensive to fix the problem by adding or re-positioning cameras in a trial and error approach.

[0006] Another approach to designing a camera layout is to visit the location, and to gain access to each candidate site for a camera. A user holds up a calibrated, rectangular device and the scene is observed through the rectangle. The size of the rectangle is varied to simulate variation in the field of view of the camera. However, this approach requires the user to visit the location and repeat the process of calibrating the rectangular device for each camera location. This approach is expensive and impractical in many locations.

[0007] With each of the above-described approaches a user is unable to ascertain an accurate indication of the sensor coverage of a desired site until all of the sensors are installed and evaluated. Therefore, there is a need in the art for a method and apparatus for dynamically providing sensor and/or camera placement using a 3D model.

SUMMARY OF THE INVENTION

[0008] In one embodiment, the present invention discloses a method and apparatus for modeling placement of sensors using a 3D model. This novel system is for displaying and interactively modifying the location, orientation, field of view, and other parameters of one or more sensors, e.g., infrared sensors, ultrasonic sensors, motion sensors, and cameras, using a 3D model of the scene. This unique approach can optionally display regions of occlusion, size of objects in the field, the image captured from each camera sensor, regions that are within the range of alarms and other parameters based on the sensors that are deployed. This novel system allows a user to drag and drop sensors on the 3D model and to adjust them interactively in order to optimize the placement of sensors at a scene.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] So that the manner in which the above recited features of the present invention can be understood in detail, a more particular description of the invention, briefly summarized above, may be had by reference to embodiments, some of which are illustrated in the appended drawings. It is to be noted,

however, that the appended drawings illustrate only typical embodiments of this invention and are therefore not to be considered limiting of its scope, for the invention may admit to other equally effective embodiments.

[0010] Figure 1 depicts a high level block diagram of a system in accordance with the invention;

[0011] Figure 2 depicts a viewpoint of a 3D model of a scene having a sensor in accordance with an embodiment of the invention;

[0012] Figure 3 depicts a viewpoint of 3D model of a scene having multiple sensors in accordance with an embodiment of the invention;

[0013] Figure 4 depicts viewpoint of a 3D model of a scene in accordance with an aspect of the invention;

[0014] Figure 5 depicts another viewpoint of the 3D model of the scene of Figure 4;

[0015] Figure 6 depicts an illustrative scene demonstrating a relationship between an occluded object and an un-occluded object in accordance with an embodiment of the invention;

[0016] Figure 7 depicts a flow chart of a method in accordance with an embodiment of the invention; and

[0017] Figure 8 depicts a high level block diagram of a computer architecture for performing an embodiment of the invention.

[0018] To facilitate understanding, identical reference numerals have been used, wherever possible, to designate identical elements that are common to the figures.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0019] The present invention is a method and apparatus for sensor placement which displays and interactively modifies the location, orientation, field of view, and other parameters of one or more sensors, e.g., cameras, infrared sensors, ultrasonic sensors, and motion sensors, using a 3D model of the scene. In addition, the method and apparatus interactively display regions of occlusion, size of objects in the field of view, the image captured from the camera sensor, regions that are within the range of alarms (video-based or otherwise), and

other image parameters in order to optimize the design of a sensor layout at a scene.

[0020] Figure 1 depicts a high level block diagram of a sensor placement system 100 in accordance with an embodiment of the invention. The system comprises a sensor placement module 101, one or more I/O devices 114, and a display 112. The system utilizes one or more I/O devices 114 for the input of information and/or adjustment of parameters by a user. The I/O devices 114 can be a keyboard, mouse, joystick, trackball or the like.

[0021] An interactive graphical user interface ("GUI") 102 receives instructions from the I/O devices 114 and selects and loads one or more pre-stored 3D models 108. In addition, the GUI 102 is also used to place sensors and to select their configuration or sensor parameters 106 such as field of view and resolution. The GUI can be used to adjust the sensor(s) and to select a viewpoint 104 from which the 3D site model 108 and the sensor coverage area are observed. The viewpoint 104, sensor parameters 106, and 3D model are rendered using a rendering module or engine 110 discussed in further detail below. As the user moves the sensor and/or adjusts the sensor parameters 106 (e.g., the position of the sensor), the user is able to dynamically observe the changes on the visual display 112. As the user adjusts the sensor parameters 106, the user is able to immediately assess the effectiveness of the placement of the sensors.

[0022] The rendering module receives the 3D site model and the sensors(s) to illustrate a scene that may show a) the correct view of the 3D scene from an arbitrary viewpoint, accounting for occluding surfaces, and b) the correct field of coverage of the sensor, given sensor parameters, while also accounting for occluding surfaces in the 3D scene. As a result, various viewing angles of the scene can be selected. The viewing angles can be selected from the perspective of a virtual sensor or from an arbitrary view point in the scene. Thus, the system 100 easily displays the effect of occlusions and, thereby allows interactive adjustments of the sensor parameters 106 and/or the addition of more sensors to minimize the effects of occlusions. As a result, the user can rapidly position cameras or other sensors in a very complex scene to assess the effectiveness and/or the coverage of the sensor placement.

[0023] The GUI 102 can be used to insert objects into a scene and to allow viewing of the scene and the camera from a different view, (i.e., from a third party perspective). The view of the scene can be selected by the user or set by default.

[0024] In the manual insertion process, the user inserts sensors into a scene and adjusts the sensor parameters of the camera in accordance with criteria desired by the user. For example, the user may position the camera to minimize occluded area or to maximize the coverage area of camera. Specifically, the user can drag-and-drop the sensor (e.g., a camera) on the 3D model and adjust it interactively in order to optimize the design of a sensor or camera layout at a scene.

[0025] In addition, the sensor placement module 101 can automatically position sensors within a 3D model and render the composite image accordingly. For example, a video can be created of a site for offline testing of algorithms with choreographed moving or static objects captured from moving or static cameras. The user specifies areas to be covered and areas of installation in the site by using the GUI to select these areas. The user also uses the GUI to answer a set of predetermined questions to identify the hardware and software configuration for the specific need(s). For example, in a typical session, the user selects the number of cameras to be used and the types of automatic detection algorithms to be applied (e.g., left-object detection, motion detection, perimeter breach alarm, and tracking). The sensor placement module 101 suggests items, including but not limited to, where the cameras are mounted in the regions specified and the orientation of each camera. The sensor placement module 101 can maximize the coverage of the critical areas and satisfy the criteria of detection algorithms. The sensor placement module 101 uses a global optimization routine to mutually adjust parameters in the system. If the sensor placement module 101 can not converge to a solution, the sensor placement module 101 suggests changing some of the parameters (e.g., increasing the camera number, etc).

[0026] The 3D site model 108 may be acquired from various sources and by various means. For example, 3D models in different 3D model formats can be used in accordance with the invention (e.g., the VRML and OBJ formats).

Commonly owned and assigned U.S. Application Serial No. 10/191,397 filed July 8, 2002 discloses an exemplary method for acquiring 3D models and is incorporated by reference herein.

[0027] However, for some applications, a 3D model of the site may not be directly available or necessary. In addition, creation of the 3D model of a site may be time consuming and expensive to generate. In these cases, the sensor placement design system 100 utilizes a library of standard 3D models, such as walls, hangars, fences, that the user can load into the system 100 and placed with respect to each other to create a full 3D model. These model modules can be pieced together to model the entire 3D scene, or just 1 or 2 of the modules can be loaded into the system 100 to model particular locations of the scene. For example, if a sensor system is designed to protect the outer perimeter of a building, then there is no need to load up models of the building, but instead, the system 100 may only need a model of the perimeter. Also, if the scene is particularly simple, such as a straight fence segment, then a model of a fence can be loaded and used to design the sensor placement.

[0028] Figure 2 depicts a viewpoint 200 of a building 201 rendered in a 3D model, in accordance with one embodiment of the invention. The building 201 has a wall 202 having a sensor 204 thereon. For illustrative purposes only, the sensor 204 is depicted as a camera. However it is appreciated that the sensor may be any sensory device (e.g., a motion detector).

[0029] Figure 2 also depicts a viewpoint (i.e., a third party perspective) of the camera 204; the area covered 206 by the camera 204; and a projected test pattern 208. The projected test pattern 208 is depicted by a checkered area which corresponds to regions observed or covered by the camera. The projected test pattern 208 is an indicator to quantify the extent of coverage (e.g., the sensitivity or resolution) for a given sensor arrangement, e.g., the smaller the checkered pattern, the greater the resolution. As the user repositions the camera 204, the area or volume of space covered 206 and the projected test pattern 208 change dynamically. This allows the user to have a sense of the total coverage area by navigating to an appropriate viewpoint within Figure 2. For example, if a user requires a certain minimum resolution (i.e., the number of pixels which cover a given area) the user can reposition the angle of the camera

204 with respect to the earth. As a result, the size of the checkered pattern changes dynamically in accordance with the movement of the camera.

[0030] Figure 2 also depicts an optional compass 212 and legend 210. The compass 212 provides a rough estimate of the orientation of the scene and of the 3D site model. The legend 210 provides a plurality of sensor parameters such as for example x, y, and z position coordinates of the camera and camera constraints (e.g., camera model limitations). As the camera 204 is moved the information (e.g., coordinates of the camera 204) in the legend 210 changes accordingly. The GUI 102 allows the user to enter a command/select an option to make legend 210 viewable. The parameters listed in the legend are sensor specific. For example, when the sensor is a camera, the legend can include such items as the cartesian coordinates (x,y,z) of the camera, orientation(pitch, yaw and roll angles) of the camera, focal length, image size, etc. When the sensor is an audio sensor, the legend can include such items as the position, orientation, and signal to noise ratio ("S/N") of the audio sensor. The S/N can be dependent upon surrounding material characteristics that have been simulated by the 3D model.

[0031] In addition, the system 100 can use a variety of different sensors 204 in a rendered 3D image for viewing on display 112. Further, the user can adjust the sensor parameters 106 for each individual sensor 204.

[0032] For clarity, Figure 2 only depicts one sensor 204 and no occlusions. However, there are instances when more than one sensor 204 is needed and occlusions are present. A detailed explanation of how the invention handles occlusions is provided in greater detail below.

[0033] Figure 3 depicts a viewpoint of a 3D image 300 having multiple sensors in accordance with an embodiment of the invention. The building 201 has multiple walls 202₁ and 202₂ (although the building 201 contains more than two walls, for clarity the building 201 is only described with respect to walls 202₁ and 202₂). Wall 202₁ has sensors 204₂, 204₃, and 204₄ thereon; and wall 202₂ has sensor 204₁ thereon (collectively sensors 204). Each of the sensors 204 in Figure 3 is depicted as a camera. However, it is appreciated that the sensors 204 need not be of the same type nor is it necessary for the sensors 204 to have the same sensor parameters 106.

[0034] As described in Figure 2, each of the cameras 204 has associated therewith a respective coverage area 206. For example, cameras 204₁, 204₂, 204₃, and 204₄ have a respective cover area 206₁, 206₂, 206₃, and 206₄. Illustratively, the coverage area of each of the cameras 204 is depicted with the test pattern 208. However, the GUI 102 can be used to optionally deselect viewing of respective test patterns 208 associated with any (or all) of the cameras 204 and the respective coverage areas 206. Figure 3 also depicts the optional compass 212 and legend 210.

[0035] In addition, Figure 3 contains frames 302 and 304. Each of the frames 302 and 304 depicts a "birds eye view" of the building 201 and the coverage provided by sensors 204. Frame 302 depicts a "bird's eye view" having a different zoom ratio than the zoom ratio of the "bird's eye view" depicted in frame 304. These "bird's eye views" allow a user to quickly assess the overall coverage given a particular placement of the sensors 204. However, these views only provide coverage information from one perspective. As discussed below other perspectives will further assist the user to ensure that the overall coverage will satisfy the requirements of a particular sensor deployment. The GUI 102 may be used to select or to de-select an option which allows viewing or hiding either or both of the frames 302 and 304.

[0036] After satisfactory placement of any or all of the cameras 204, a report can be generated. Illustratively, the report can contain such items as a view of where the cameras are placed on the wall 204 and the orientation of the cameras; lists all the chosen camera parameters; and images that show the coverage and occluded (i.e., blind spots) areas of the cameras.

[0037] Often an occlusion is present which blocks all or part of the coverage area 206 of a sensor 204. Figure 4 depicts a viewpoint of a 3D model 400 having a sensor 402, illustratively a camera, with an occluded area. Figure 4 illustrates a building 201, a connecting corridor 406, compass 212, and a camera 402. The camera 402 is mounted to one of the multiple walls which form the building 201. Although the building 201 has multiple walls, the invention is described with respect to only the wall 202 which has the camera 402 mounted to it.

[0038] The camera 402 has a coverage area 206, an occluded area 404, and a projected test pattern 208 within the coverage area 206. As indicated above, the projected test pattern 208 is depicted as a checkered area that acts as an indicator to quantify the extent of coverage for a given sensor arrangement. Note that the occlude region 404 is present within the covered area 206 but is not part of the projected test pattern 208. The rendering module 110 is able to determine that an occlusion area 404 is present within the coverage area 206 of the camera 402. A method for computing the occluded area 404 is discussed in greater detail below.

[0039] In one embodiment, the user can interactively adjust the parameters of the camera 402 using the I/O devices 114. As a result, the effect of the adjustments can be observed immediately. For example, a user may wish to minimize the occluded area 404. The user can reposition the camera on the wall 202 and/or adjust an angle of the camera with respect to the wall 202. The user adjustments are communicated to the rendering module 110. Once the desired result is achieved, additional sensors (e.g., cameras or ultrasonic sensors) can be placed into the 3D model. The projected test patterns 208 for multiple cameras can be simultaneously shown to help ensure that the cameras 402 are optimally placed.

[0040] Figure 5 depicts a different viewpoint 500 of the 3D model of Figure 4. As such, the same sensor parameters 106 are present for the camera in the 3D model of Figures 4 and 5. In addition to the elements discussed with respect to Figure 4, Figure 5 further comprises a camera perception 504 and objects within the occluded area 404. Illustratively, the objects are people 502₁, 502₂, and 502₃ (collectively people 502).

[0041] The camera perspective 504 allows a user to view what the camera 402 sees within the coverage area 206 using the current sensor parameters 106. The camera perspective 504 demonstrates that while located within occluded area 404 the people 502 are not viewable from the current position of the camera 402. This viewing capability will quickly allow a user to detect gaps in the coverage of the sensors. Further, the camera perspective 504 is an optional feature which may be selected by the user via the GUI 102.

[0042] To determine when an occlusion is present, the system 100 uses a rendering algorithm. Examples of rendering methods that can be used with the present invention are disclosed in commonly assigned U.S. Patent Application Serial Nos. 10/202,546, filed July 24, 2002 and 09/800,550, filed March 7, 2001. The material contained in the '546 and '550 applications is incorporated by reference herein. It is appreciated that various rendering algorithms can be used in accordance with the invention and that the algorithms disclosed herein are for exemplary purposes and not intended in any way to limit the scope of the invention.

[0043] For illustrative purposes, the rendering algorithm disclosed in the U.S. Patent Application Serial No. 10/202,546 is briefly described herein and is referred to as the "VIDEO FLASHLIGHTS" algorithm. The VIDEO FLASHLIGHT algorithm generates an image at each time instant from a virtual camera viewpoint specified by the user through the GUI 102. Using, VIDEO FLASHLIGHT, an image consists of pixels that show either a test pattern video or an actual video in regions that are illuminated by any of the real cameras, and pixels that contain background model textures for regions that are not illuminated by any camera. The background model textures are static and are represented as standard 3D graphics model representations along with the 3D vertices of the model triangles and their texture coordinates. The textures from the live/test video frames change at the video frame rate. The composite image created for any virtual viewpoint combines the two in the rendering implementation.

[0044] When rendering using VIDEO FLASHLIGHT, the light source used in the shadow map algorithm is replaced by a real video flash light camera. The video texture for every frame is considered the "illuminant". The invention implements shadow mapping by taking advantage of projective texturing, depth buffering, automatic texture generation, texture compositing and alpha testing. On platforms that support multi-texturing, this method generally takes one rendering pass for shadow map generation.

[0045] Figure 6 depicts an illustrative scene 600 demonstrating a relationship between objects O1 606 and O2 608. Figure 6 comprises objects O1 606 and O2 608, a viewpoint of a real flashlight camera 612 (i.e., the VIDEO

FLASHLIGHT camera), and a viewpoint of a user-specified virtual camera 610. This relationship is useful in explaining how the sensor placement module 101 detects occlusions. For example, the virtual camera 610 sees both objects O1 606 and O2 608. However, the flashlight camera 612 sees object O1 606 only. Thus O2 608 is occluded by O1 606 from the viewpoint of the flashlight camera 612. As a result, only O1 606 is illuminated by textures from the flashlight camera 612. O2 608 comprises static model textures which represent the 3D model site. Thus, when rendering the virtual camera view 610, only the pixels seen by the flashlight camera 612 and the virtual camera 610 are projected with the dynamic video textures.

[0046] Specifically, the invention associates two depth values for each point in the scene. The first depth value corresponds to a z-buffer depth for the flashlight camera 612. Each scene that points along a view ray from the flashlight camera is assigned a depth value corresponding to that ray's z-buffer value (e.g., a projective texture). The second depth value is the true depth value for each scene point with respect to the flashlight camera 612. By comparing these two values for every point in the view of the virtual camera 610, it can be established whether the point is seen by the flashlight camera 612 or is occluded. Points with identical depth values are seen by the flashlight camera 612 and should be rendered with the flashlight video texture.

[0047] The occlusion handling technique can be implemented as a multi-pass rendering algorithm. The scene 600 is first rendered in the flashlight camera coordinate system to obtain the z-buffer. The z-buffer is stored in the alpha channel of the texture used for flashlight video projection. This is the first alpha value and for each scene point represents the depth of the first hit for a view ray from that point to the flashlight camera center. A second rendering uses automatic texture coordinate generation to set the texture coordinate for each vertex to the true depth value for the vertex with respect to the flashlight camera 612. A 1D ramp-texture is used to define a mapping of this depth to alpha values. This is the second alpha value and represents the true depth for each scene vertex in the flashlight camera coordinates. As a result, for all points visible from the virtual camera 610, there are two alpha values that are compared to determine which points are illuminated by the flashlight camera.

Pixels that satisfy the equality test for the alpha values are illuminated by the flashlight camera 612 and are rendered with the flashlight camera texture while the rest are rendered with the model texture.

[0048] For illustrative purposes, an example of pseudo-code for a rendering algorithm is provided:

```
Display {
  for all visible flashlight cameras {
    if (play)
      UpdateVideoContent(Video Source, Frame Number);
    if (moving)
      UpdateDepthMap();
    else
      UpdateDepthMapOnce();
  }
  SetupViewport(ScreenResolution);
  ClearScreenColorBuffer();
  ClearScreenDepthBuffer();
  MultMatrix(Inverse(Virtual Camera Pose));
  RenderScene(Geometry+Textures);
  for all visible flashlight cameras
    ExtractVisiblePixels();
}
UpdateDepthMap {
  SetupViewport(DepthMapResolution);
  ClearDepthBuffer();
  MultMatrix(Inverse(Camera Pose));
  // Decrease precision error by offsetting the geometry
  SetPolygonOffset;
  MaskColors; // Only need z-buffer
  RenderScene(Geometry);
  ReadDepthBuffer();
  TransferDepthToAlphaChannel(VideoTextureRGBA);
}
ExtractVisiblePixels {
  SetupTexture1();

  BindTexture(RampTexture);
  // ZToTextureCoordinateMatrix extracts
  // Z-component
  SetupEyeLinearTextureCoordinateGeneration(
  ZToTextureCoordinateMatrix*Inverse(Camera Pose));
  SetupTextureEnvironment(UseTextureAlpha);
  SetupTexture2();
  BindTexture(VideoTextureRGBA);
  SetupEyeLinearTextureCoordinateGeneration(
  Inverse(Camera Pose));
  SetupTextureEnvironment(
```

```
SubtractPreviousTextureAlpha,UseTextureRGB);  
EnableAlphaTesting();  
SetAlphaFunc();  
// Avoid back projection  
SetupCullingPlane(CameraImagePlane);  
RenderScene(Geometry);  
}
```

[0049] In the algorithm, video content and depth maps are updated on an as-needed basis. Depth map textures are obtained by rendering the geometry in either the frame-buffer or in the p-buffer. During this process, polygon offsetting is utilized in order to avoid re-sampling and precision errors. Once all the textures are updated, a scene is rendered with the static model (background) textures. Then for visible flashlight cameras, two textures are projected using the corresponding camera pose matrices.

[0050] The first texture uses an extra matrix operation ("ZToTextureCoordinateMatrix") to map the Z values of all the scene points in the frustum of a flashlight camera in the camera coordinates to the s coordinate in the texture coordinates. Because the occlusion calculation is performed in the texture domain, the Z values are mapped to texture values using a 1D ramp texture. The Z-values corresponding to the scene points behind the camera are culled by specifying the camera image plane as the culling plane.

[0051] The second texture has the z-buffer depth map in the alpha channel and the video content in the RGB channel. The texture environment is set such that the alpha values are subtracted from the previous texture's alpha, which implements the occlusion test depicted in Fig. 6.

[0052] Shadow map extensions can be used with the invention. For example, the z-buffer described above is transferred to a shadow map texture. The output values, usually 0 or 1, generated by this texture, flag the pixels as shadowed or illuminated respectively. These flags can be used for rendering the final pixels. When the automatic texture generation in the flashlight camera coordinates is turned on, for a computed texture coordinate, $\{s, t, r, q\}$, $\{s/q, t/q\}$ points to the z-buffer value, and r/q represents the true depth. Implicit comparison of the z-buffer value with the true depth enables occlusion handling in the shadow map extensions.

[0053] Figure 7 depicts a flowchart of a method 700 in accordance with the invention. The method 700, begins at step 702 and proceeds to step 704. At step 704, an interface (e.g., a GUI) is used to select a sensor. Further, the GUI 102 is used to position the sensor 204 (e.g., a camera) onto a selected 3D site model. In one embodiment, the 3D site model is a model of an actual site where the sensor(s) are subsequently installed. In another embodiment, a user can use the GUI 102 to select a generic model suitable for use in determining where sensor(s) are to be placed at a location (e.g., a model of a fence or wall).

[0054] At step 706, sensory parameters 106 associated with a sensor, the 3D site model, and a viewpoint 104 are rendered using the 3D rendering algorithm. Specifically, the 3D rendering module 110 uses a rendering algorithm to compare, on a pixel by pixel basis, the pixels observable from the viewpoint of the camera and the pixels observable from a user selected virtual camera. When the pixels are different, the method 700 determines that an occlusion is present. When a user adjusts the sensory parameters using the sensory characteristic module 106 the rendered 3D image changes accordingly. The method 700 proceeds to and ends at step 708.

[0055] In yet another embodiment, the user uses the GUI 102 to enter the number of sensors and/or sensor characteristic 106 for the selected 3D site mode. The rendering module 110 can automatically position the sensor(s) in accordance with constraints determined by the user. For example, if the user selects a number of sensors for placement in a 3D site model, the rendering module 110 can place the sensors to maximize the coverage area (e.g., by minimizing the occluded areas).

[0056] Figure 8 depicts a general purpose computer 800 suitable for use in performing the method of Figure 7. The general purpose computer of Figure 8 comprises a processor 810 as well as a memory 804 for storing control programs and the like. The processor 810 cooperates with conventional support circuitry 808 such as power supplies, clock circuits, cache memory and the like as well as circuits that assist in executing the software routines 806 stored in the memory 804. As such, it is contemplated that some of the process steps discussed herein as software processes may be loaded from a storage device (e.g., an optical drive, floppy drive, disk drive, etc.) and implemented

within the memory 804 and operated by the processor 810. Thus, various steps and methods of the present invention can be stored on a computer readable medium. The general purpose computer 800 also contains input-output circuitry 802 that forms an interface between the various functional elements communicating with the general purpose computer 800. For example, in the embodiment of FIG. 8, the general purpose computer 800 communicates with I/O devices 114. The processor 810 interprets inputs received from the I/O devices 114 and, in response thereto, the processor 810 forwards the inputs to the sensor placement module 101. The sensor placement module 101 uses the instructions (e.g., the relating to the viewpoint 104 (either default or selected), the sensor parameters 106, and the 3D site model 108) to render a 3D image.

[0057] Although Figure. 8 is depicted as a general purpose computer that is programmed to perform various control functions in accordance with the present invention, the invention can be implemented in hardware, for example, as an application specified integrated circuit (ASIC). As such, the process steps described herein are intended to be broadly interpreted as being equivalently performed by software, hardware, or a combination thereof.

[0058] In one embodiment of the invention, the viewpoint 104 of a 3D site model and/or the sensory parameters 106 for a sensor is set by default. In other embodiments, the user uses the GUI 102 to select a viewpoint 104 and/or the sensory parameters 106 of the sensor.

[0059] Furthermore, the invention disclosed herein can use shadow mapping, clipping planes, or special texture culling operations depending on, for example, the hardware used and the level of performance desired. Shadow mapping utilizes texturing, depth buffering, and arithmetic/logical operations at the texture and fragment levels. For example, hardware can be used for shadow mapping which employs projective textures, texture compositing and fragment testing without the need to rely on any restrictive hardware extension. In other embodiments, dedicated extensions such as SGI extensions can be used.

[0060] While the foregoing is directed to embodiments of the present invention, other and further embodiments of the invention may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.